🎓 MLOps Assignment 3: NASA APOD Data Pipeline

Deadline Nov 16, 2025

This assignment serves as a critical capstone project, requiring you to integrate essential MLOps tools—**Airflow, Astronomer, DVC, and Postgres**—within a unified, containerized environment. Your primary goal is to build a robust, reproducible **Extract, Transform, Load (ETL)** pipeline.

I. ⚙️ Pipeline Workflow Tasks (The Airflow DAG)

Your Airflow Directed Acyclic Graph (DAG) must successfully execute the following five sequential steps:

- **Step 1: Data Extraction (E)**
  - Initiate a connection to the public NASA Astronomy Picture of the Day (APOD) endpoint (`https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY`) to retrieve the daily structured data.
- **Step 2: Data Transformation (T)**
  - Select specific fields of interest (e.g., date, title, URL, explanation) and restructure the raw JSON response into a clean, usable format, likely leveraging a Pandas DataFrame.
- **Step 3: Data Loading (L)**
  - Simultaneously persist the cleaned data to two distinct storage locations:
    - An existing table in your **PostgreSQL database**.
    - A local **CSV file** (e.g., `apod_data.csv`) within the Airflow environment.
- **Step 4: Data Versioning (DVC)**
  - Execute appropriate DVC commands within the pipeline to place the newly created **CSV file** under version control, creating a corresponding metadata file (e.g., `apod_data.csv.dvc`).
- **Step 5: Code Versioning (Git/GitHub)**
  - Perform a Git operation to commit the updated **DVC metadata file** (`.dvc`) to your main GitHub repository, linking the pipeline code to the exact version of the data it produced.

II. 💡 Conclusion, Objective, and Key Learnings

- **Primary Objective:** To successfully design, implement, and deploy a **reproducible MLOps data ingestion pipeline** using **Docker/Astronomer** for deployment parity.
- **Key Learning Outcomes:**
    - **Orchestration Mastery:** Gain practical experience defining complex, dependent workflows using **Apache Airflow**.
    - **Data Integrity:** Understand the trade-offs and techniques for concurrent loading of data into both a relational database (**Postgres**) and a file storage system (CSV).
    - **Data Lineage:** Master the use of **DVC** alongside **Git** to ensure that all data artifacts are versioned and traceable, which is fundamental for model reproducibility.
    - **Containerized Deployment:** Demonstrate proficiency in preparing a custom Docker image to include all necessary libraries (e.g., DVC, `psycopg2`) for deployment to platforms like **Astronomer**.

The End 🙂